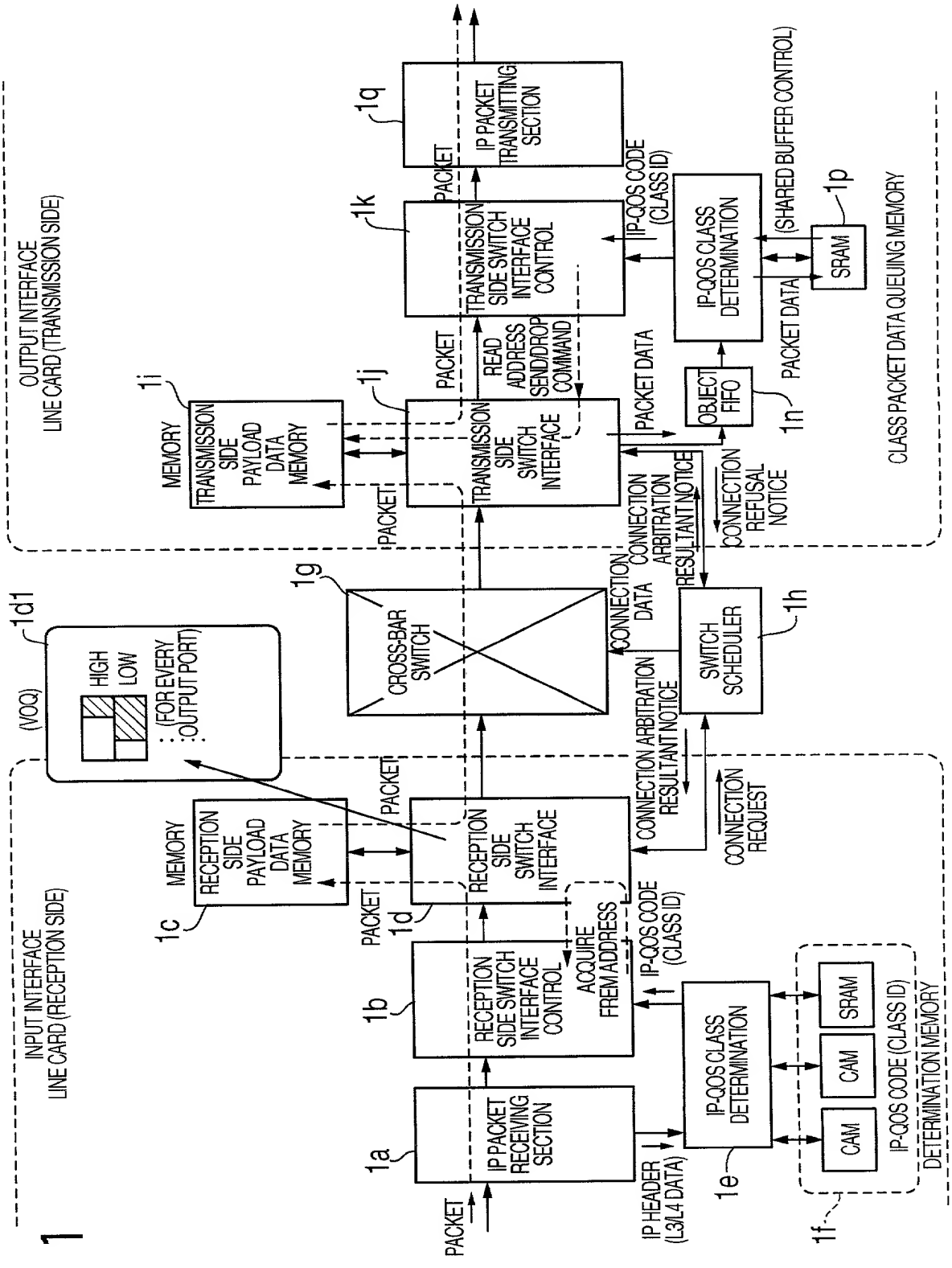

INPUT INTERFACE
LINE CARD (RECEPTION SIDE)
OUTPUT INTERFACE
LINE CARD (TRANSMISSION SIDE)
CLASS PACKET DATA QUEUING MEMORY
IP PACKET RECEIVING SECTION
RECEPTION SIDE SWITCH INTERFACE CONTROL
ACQUIRE FREM ADDRESS
IP-QOS CLASS DETERMINATION
CAM
CAM
SRAM
IP-QOS CODE (CLASS ID)
DETERMINATION MEMORY
MEMORY
RECEPTION SIDE PAYLOAD DATA MEMORY
RECEPTION SIDE SWITCH INTERFACE
(VOQ)
HIGH
LOW
: (FOR EVERY OUTPUT PORT)
CROSS-BAR SWITCH
SWITCH SCHEDULER
CONNECTION DATA
CONNECTION ARBITRATION RESULTANT NOTICE
CONNECTION REQUEST
CONNECTION REFUSAL NOTICE
TRANSMISSION SIDE PAYLOAD DATA MEMORY
TRANSMISSION SIDE SWITCH INTERFACE
TRANSMISSION SIDE SWITCH INTERFACE CONTROL
IP PACKET TRANSMITTING SECTION
OBJECT FIFO
IP-QOS CLASS DETERMINATION
SRAM
(SHARED BUFFER CONTROL)
PACKET
PACKET DATA
READ ADDRESS SEND/DROP COMMAND
IP HEADER (L3/L4 DATA)
IP-QOS CODE (CLASS ID)
1a
1b
1c
1d
1d1
1e
1f
1g
1h
1i
1j
1k
1n
1p
1q

Fig. 1

Fig. 2

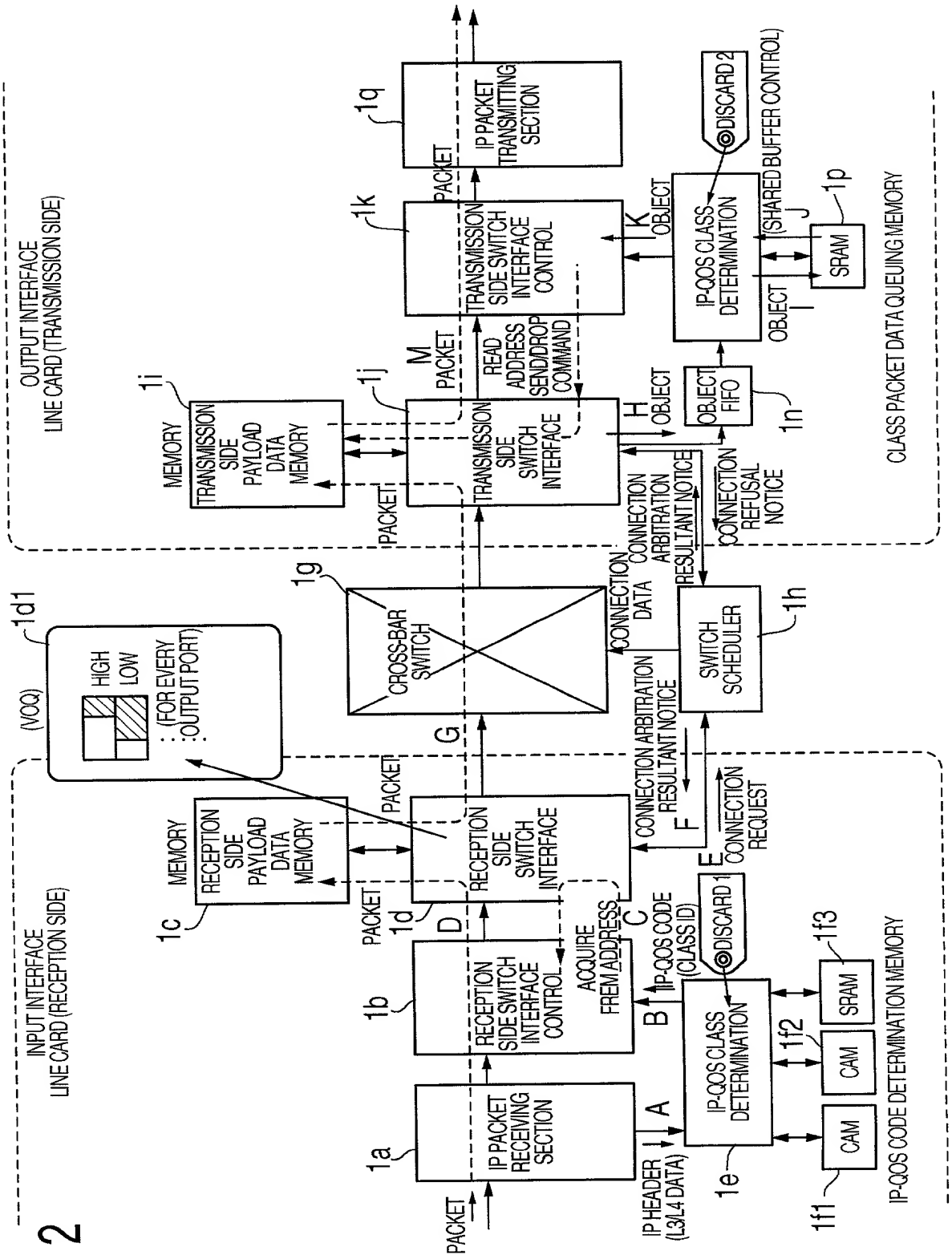


Fig. 3

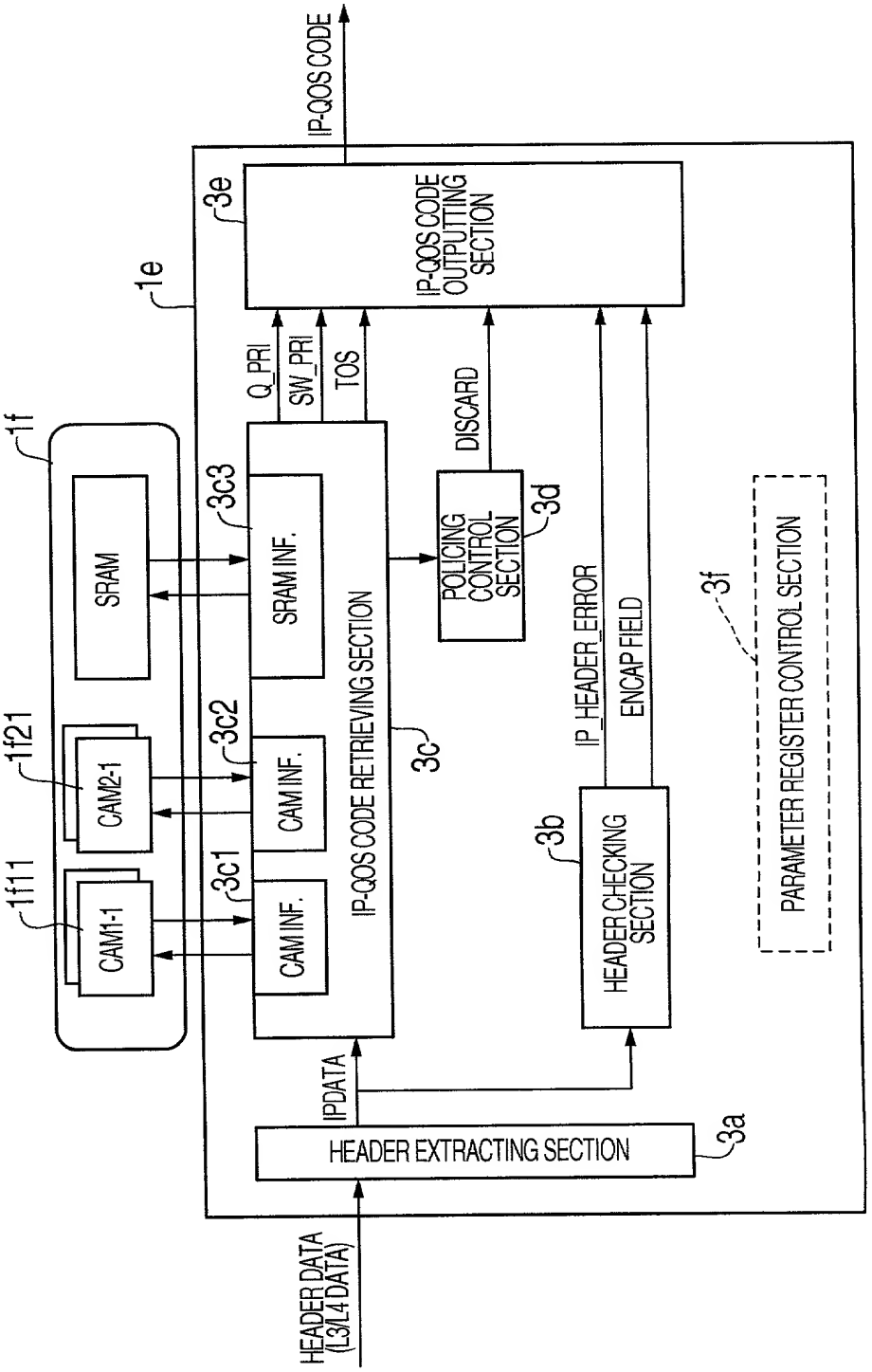
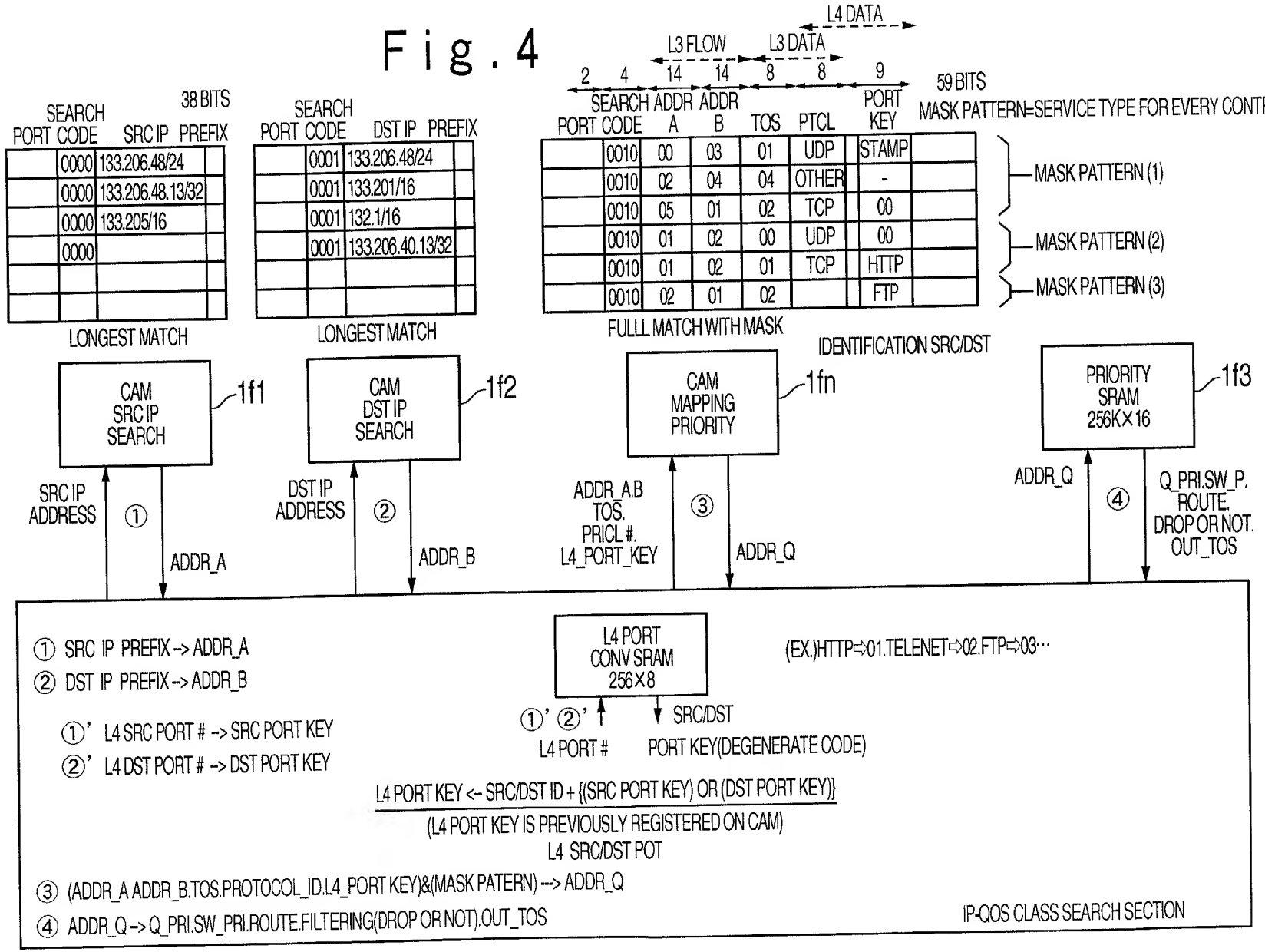

IP-QOS CODE
3e
IP-QOS CODE OUTPUTTING SECTION
1e
Q_PRI
SW_PRI
TOS
DISCARD
1f
SRAM
3c3
SRAM INF.
POLICING CONTROL SECTION
3d
1f21
CAM2-1
3c2
CAM INF.
IP-QOS CODE RETRIEVING SECTION
3c
IP_HEADER_ERROR
ENCAP FIELD
3f
1f11
CAM1-1
3c1
CAM INF.
3b
HEADER CHECKING SECTION
PARAMETER REGISTER CONTROL SECTION
IPDATA
HEADER EXTRACTING SECTION
3a
HEADER DATA (L3/L4 DATA)

Fig. 4



Michio MASUDA et al.
"Multi-Layer Class Identifying…"
Q62568
Filed January 3, 2001
Sheets 4 of 16

Fig. 5

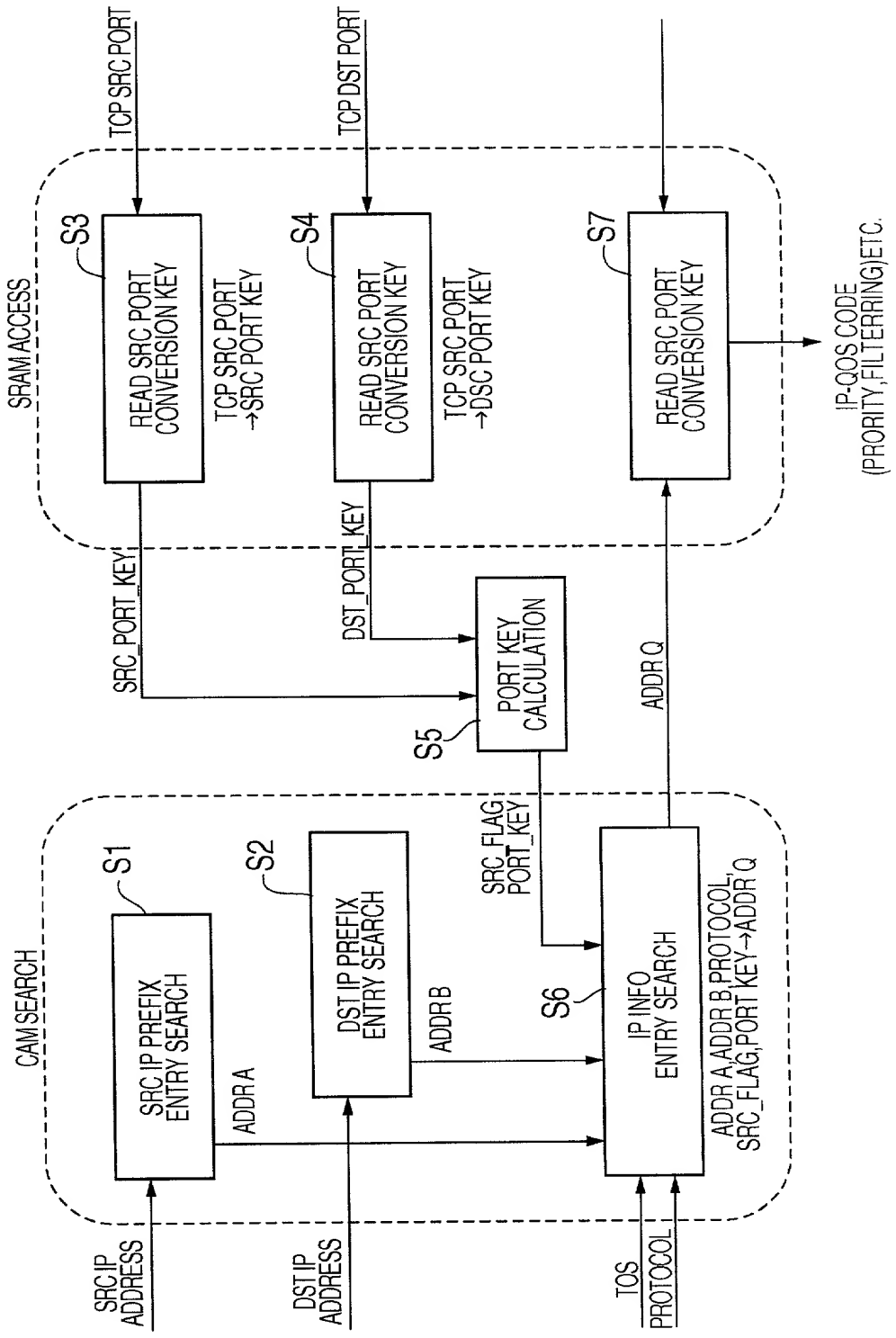

CAM SEARCH
SRAM ACCESS
SRC IP ADDRESS
DST IP ADDRESS
TOS
PROTOCOL
TCP SRC PORT
TCP DST PORT
S1
SRC IP PREFIX ENTRY SEARCH
ADDR A
S2
DST IP PREFIX ENTRY SEARCH
ADDR B
S3
READ SRC PORT CONVERSION KEY
TCP SRC PORT →SRC PORT KEY
S4
READ SRC PORT CONVERSION KEY
TCP SRC PORT →DSC PORT KEY
S5
PORT KEY CALCULATION
SRC_PORT_KEY
DST_PORT_KEY
SRC_FLAG PORT_KEY
S6
IP INFO ENTRY SEARCH
ADDR A,ADDR B,PROTOCOL, SRC_FLAG,PORT KEY→ADDR Q
S7
READ SRC PORT CONVERSION KEY
ADDR Q
IP-QOS CODE (PRORITY,FILTERRING)ETC.

Fig. 6A

【CAM REGION DIVISION】

CAM ADDRESS	CAM DATA (MAX.64 BITS)	MASK PATTERN (64 BITS)	SEARCH METHOD
ADDR_A~	IP SRC PREFIX ENTRY STORAGE REGION		LONGEST MATCH
ADDR_B~	IP DST PREFIX ENTRY STORAGE REGION		LONGEST MATCH
ADDR_Q~	IP INFO SEARCH ENTRY STORAGE REGION		FULL MATCH WITH MASK

Fig. 6B

【1,IP SRC PREFIX ENTRY STORAGE REGION : SEARCH CODE 0000】

CAM ADDRESS (ADDR_A)	CAM DATA (38 BITS)			
	HW #(2)	SEARCH CODE(4)	IP SRC ADRESS/ PREFIX(32BITS)	NON USED(26 BITS)
A #1	00	0000	IP SRC ADDRESS #1/PREFIX	
A #2	00	0000	IP SRC ADDRESS #2/PREFIX	
A #3	01	0000	IP SRC ADDRESS #1/PREFIX	
⋮	⋮	⋮	⋮	

Fig. 7A

【2,IP DST PREFIX ENTRY STORAGE REGION : SEARCH CODE 0001】

CAM ADDRESS (ADDR_B)	CAM DATA (38 BITS)			
	HW #(2)	SEARCH CODE(4)	IP DST ADDRESS/ PREFIX(32BITS)	NON USED(26 BITS)
B #1	00	0001	IP DST ADDRESS #1/PREFIX	
B #2	00	0001	IP DST ADDRESS #2/PREFIX	
B #3	01	0001	IP DST ADDRESS #1/PREFIX	
⋮	⋮	⋮	⋮	

Fig. 7B

【3,IP INFO ADDRESS ENTRY STORAGE REGION : SEARCH CODE 0010】

CAM ADDRESS (ADDR_Q)	CAM DATA (55 BITS)								
	HW #(2)	SEARCH CODE(4)	ADDR A(14)	ADDR B(14)	TOS (8)	PROTO COL# (8)	SRC/ DST (1)	PORT KEY (8)	NON USED (5 BITS)
Q #1	00	0010	A1	B1	01	TCP	S	HTTP	
Q #2	00	0010	A1	B2	04	UDP	D	SNMP	
Q #3	01	0010	A3	B1	02	TCP	S	FTP	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Fig.8

【IP INFO ENTRY】

ADDRESS(16 BITS) : UPPER 2 BITS=00 LOWER 14 BITS=HIT ADDR_Q	DATA(24 BIT)							
	Q_PRI(4)	D	P	ROUTE(1+4)		OUTPUT TOS(2+8)		RESERVE (3)
ADDR Q0	0000	0	0	0	0000	11	011011 00	
ADDR Q1	1101	0	1	0	0000	11	011010 00	
ADDR Q2	1101	0	0	0	0000	00	000000 00	
⋮	⋮	⋮	⋮		⋮		⋮	
ADDR QI	1110	0	1	1	0101	00	000000 00	
⋮	⋮	⋮	⋮		⋮		⋮	

Fig. 9

(IPV4 & TCP/UDP/OTHER HEADER FORMAT)

WORD	63			47	31			15
-	EMPTY DATA				PPP HEADER			
0	VER	IHL	TOS	DETAGRAM LENGTH	IDENTIFICATION		M	FRAGMENT OFFSET
1	TTL		PROTOCOL	HEADER CHECKSUM	SRC IP ADDRESS			
2	DST IP ADDRESS				L4 SRC PORT			L4 DST PORT

Fig. 10

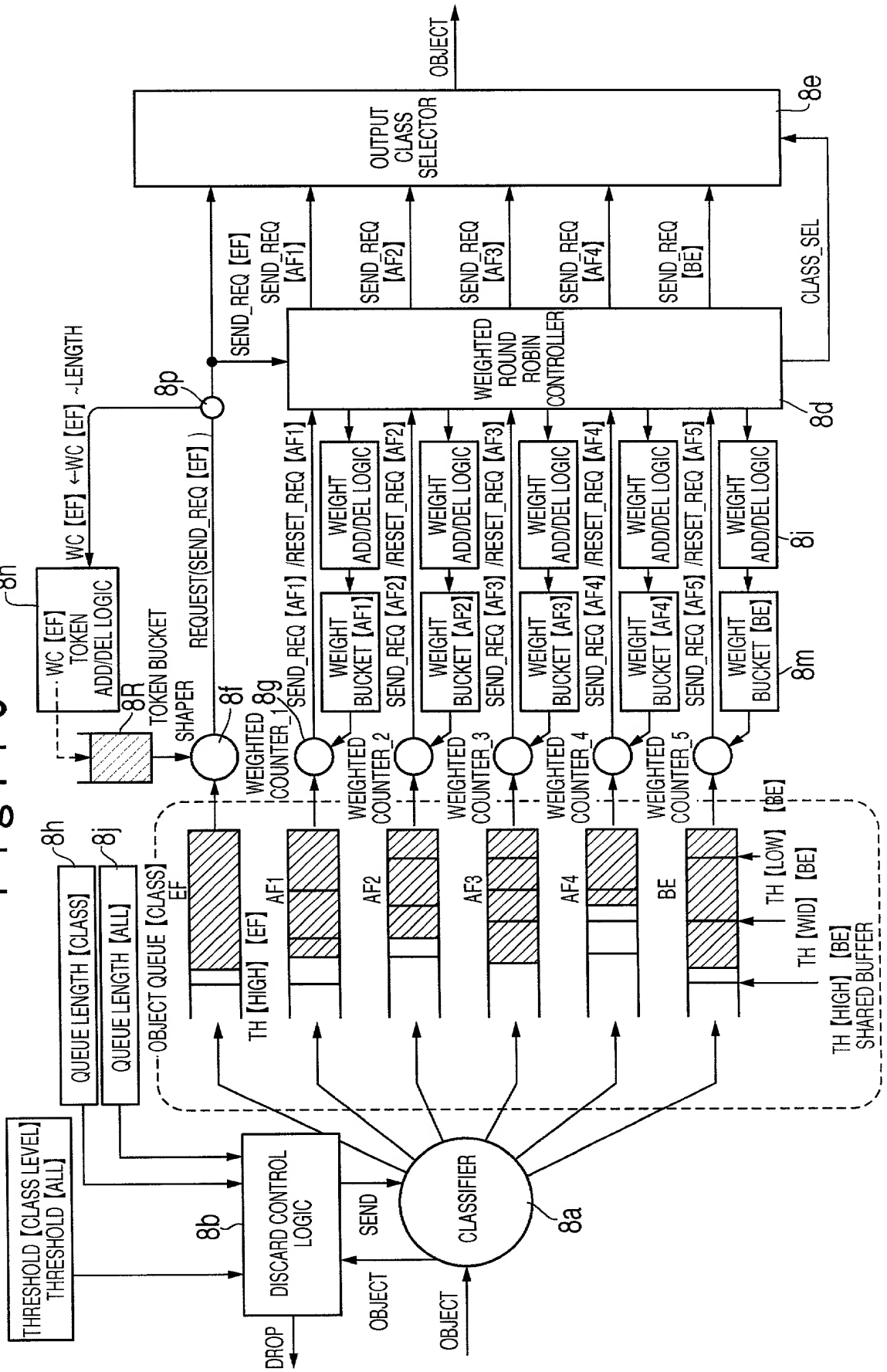

OBJECT
OUTPUT CLASS SELECTOR
8e
SEND_REQ [EF]
SEND_REQ [AF1]
SEND_REQ [AF2]
SEND_REQ [AF3]
SEND_REQ [AF4]
SEND_REQ [BE]
CLASS_SEL
WEIGHTED ROUND ROBIN CONTROLLER
8d
8p
WC [EF] ←WC [EF] -LENGTH
REQUEST(SEND_REQ [EF])
8n
WC [EF] TOKEN ADD/DEL LOGIC
8R
TOKEN BUCKET SHAPER
8f
8g
WEIGHTED COUNTER_1
WEIGHTED COUNTER_2
WEIGHTED COUNTER_3
WEIGHTED COUNTER_4
WEIGHTED COUNTER_5
SEND_REQ [AF1] /RESET_REQ [AF1]
SEND_REQ [AF2] /RESET_REQ [AF2]
SEND_REQ [AF3] /RESET_REQ [AF3]
SEND_REQ [AF4] /RESET_REQ [AF4]
SEND_REQ [AF5] /RESET_REQ [AF5]
WEIGHT ADD/DEL LOGIC
8i
WEIGHT BUCKET [AF1]
WEIGHT BUCKET [AF2]
WEIGHT BUCKET [AF3]
WEIGHT BUCKET [AF4]
WEIGHT BUCKET [BE]
8m
8h
8j
QUEUE LENGTH [CLASS]
QUEUE LENGTH [ALL]
OBJECT QUEUE [CLASS]
EF
AF1
AF2
AF3
AF4
BE
TH [HIGH] [EF]
TH [LOW] [BE]
TH [MID] [BE]
TH [HIGH] [BE]
SHARED BUFFER
8k
THRESHOLD [CLASS LEVEL]
THRESHOLD [ALL]
8b
DISCARD CONTROL LOGIC
SEND
DROP
OBJECT
CLASSIFIER
8a
OBJECT

Fig. 11

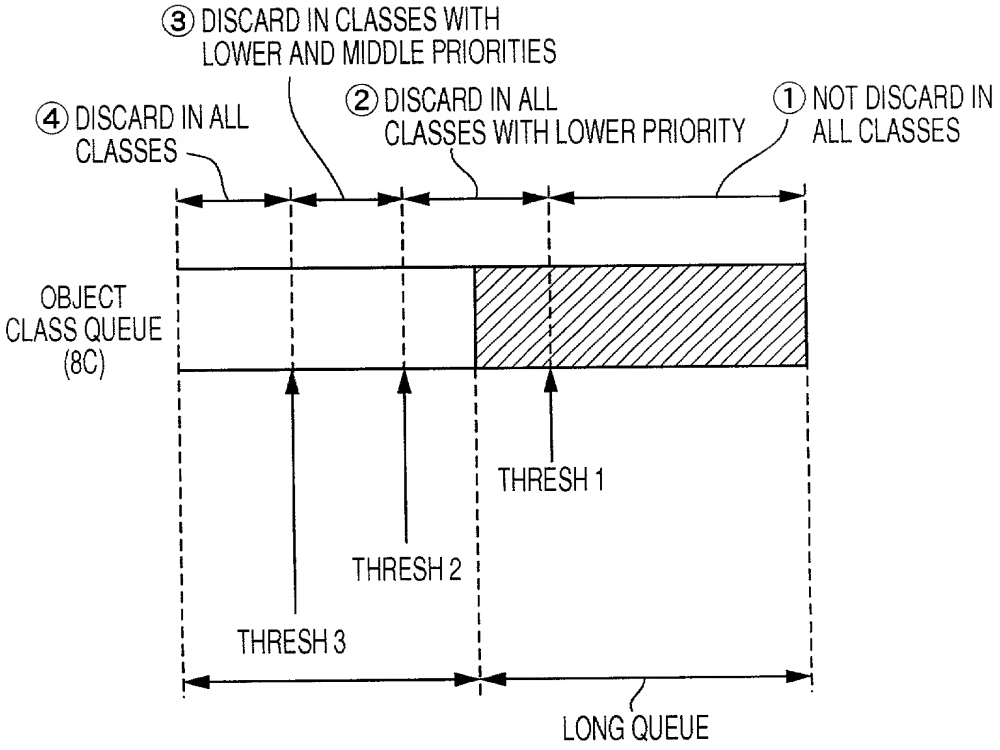

③ DISCARD IN CLASSES WITH LOWER AND MIDDLE PRIORITIES
④ DISCARD IN ALL CLASSES
② DISCARD IN ALL CLASSES WITH LOWER PRIORITY
① NOT DISCARD IN ALL CLASSES
OBJECT CLASS QUEUE (8C)
THRESH 1
THRESH 2
THRESH 3
LONG QUEUE

Fig. 12

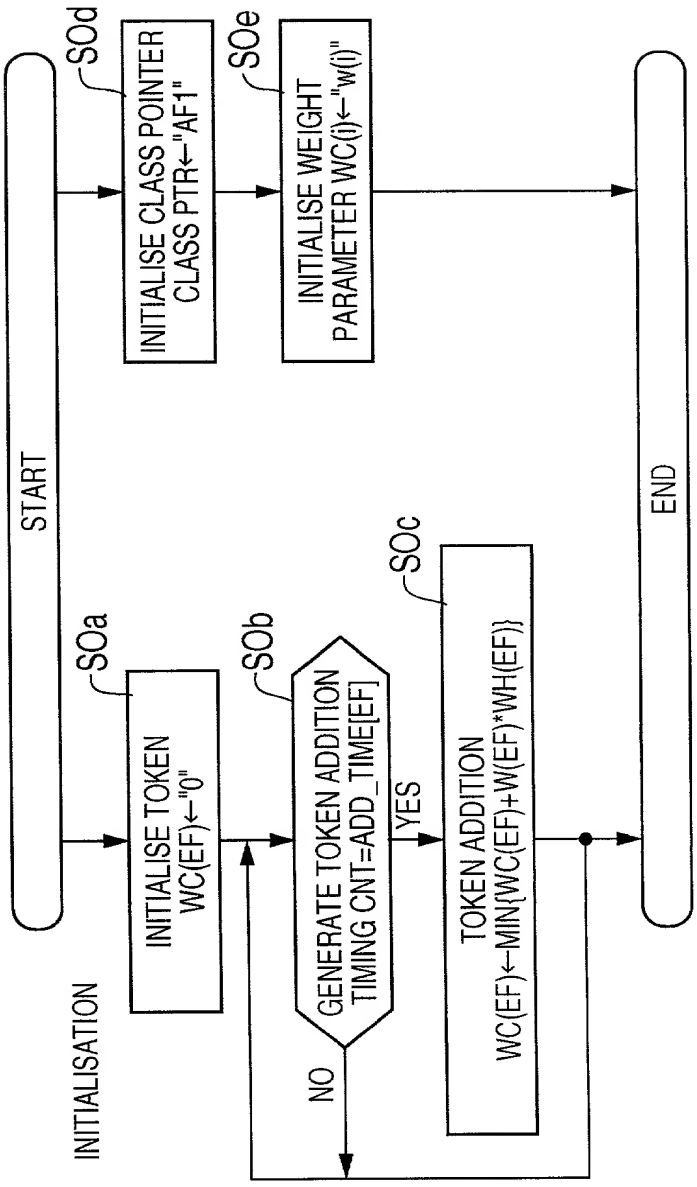
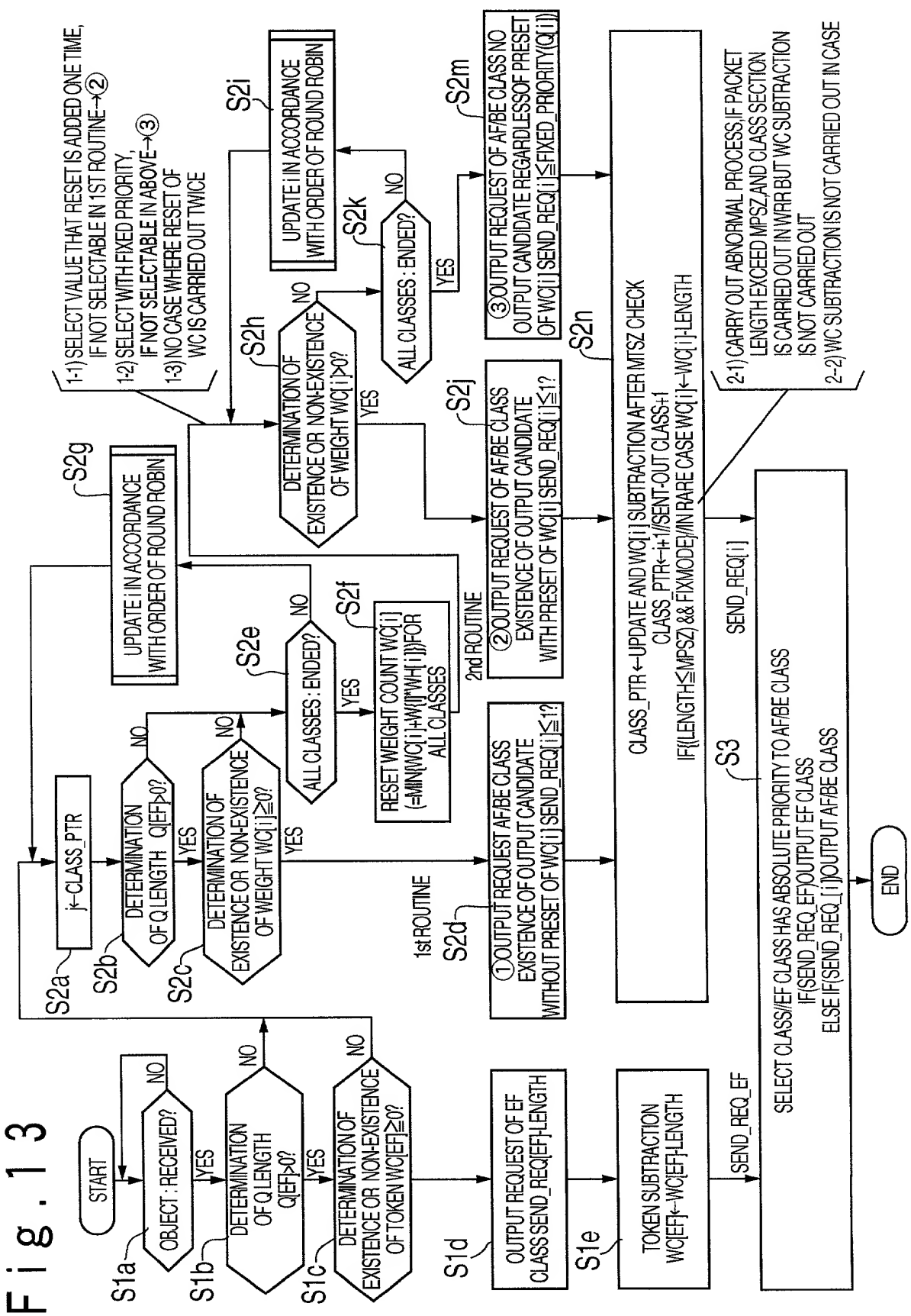

START
INITIALISATION
SOa
INITIALISE TOKEN
WC(EF)←"0"
SOb
GENERATE TOKEN ADDITION
TIMING CNT=ADD_TIME[EF]
NO
YES
SOc
TOKEN ADDITION
WC(EF)←MIN{WC(EF)+W(EF)*WH(EF)}
SOd
INITIALISE CLASS POINTER
CLASS PTR←"AF1"
SOe
INITIALISE WEIGHT
PARAMETER WC(i)←"w(i)"
END

Fig. 13



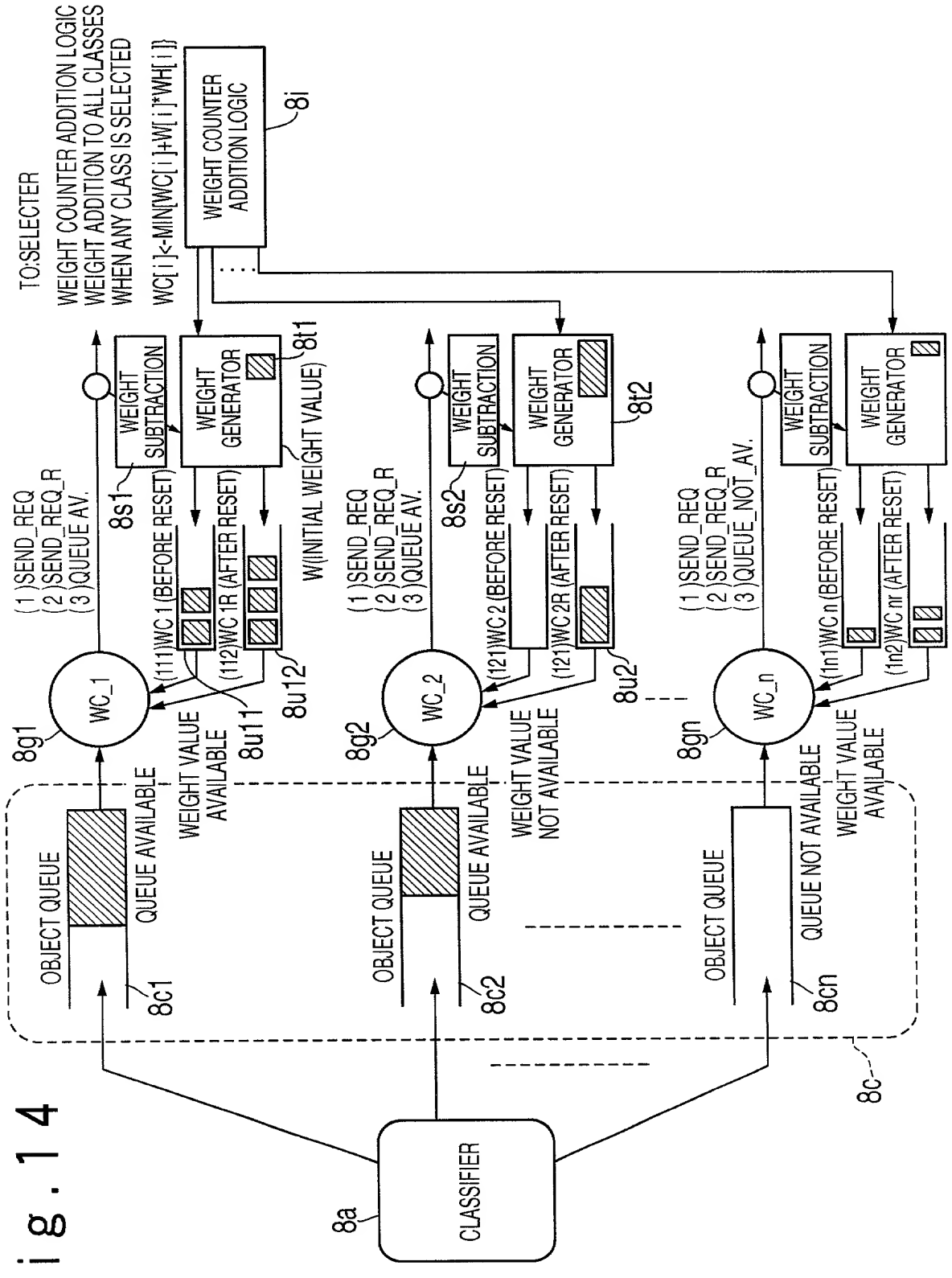

TO:SELECTER
WEIGHT COUNTER ADDITION LOGIC
WEIGHT ADDITION TO ALL CLASSES
WHEN ANY CLASS IS SELECTED
WC[i]<-MIN[WC[i]+W[i]*WH[i]]
WEIGHT COUNTER ADDITION LOGIC
8i
WEIGHT SUBTRACTION
WEIGHT GENERATOR
8t1
8s1
(1)SEND_REQ
(2)SEND_REQ_R
(3)QUEUE AV.
(111)WC 1 (BEFORE RESET)
(112)WC 1R (AFTER RESET)
W(INITIAL WEIGHT VALUE)
WC_1
8g1
8u11
8u12
WEIGHT VALUE AVAILABLE
OBJECT QUEUE
QUEUE AVAILABLE
8c1
WEIGHT SUBTRACTION
WEIGHT GENERATOR
8t2
8s2
(1)SEND_REQ
(2)SEND_REQ_R
(3)QUEUE AV.
(121)WC 2 (BEFORE RESET)
(121)WC 2R (AFTER RESET)
WC_2
8g2
8u2
WEIGHT VALUE NOT AVAILABLE
OBJECT QUEUE
QUEUE AVAILABLE
8c2
WEIGHT SUBTRACTION
WEIGHT GENERATOR
(1)SEND_REQ
(2)SEND_REQ_R
(3)QUEUE_NOT_AV.
(1n1)WC n (BEFORE RESET)
(1n2)WC nr (AFTER RESET)
WC_n
8gn
WEIGHT VALUE AVAILABLE
OBJECT QUEUE
QUEUE NOT AVAILABLE
8cn
8c
CLASSIFIER
8a

Fig. 14

Fig. 15

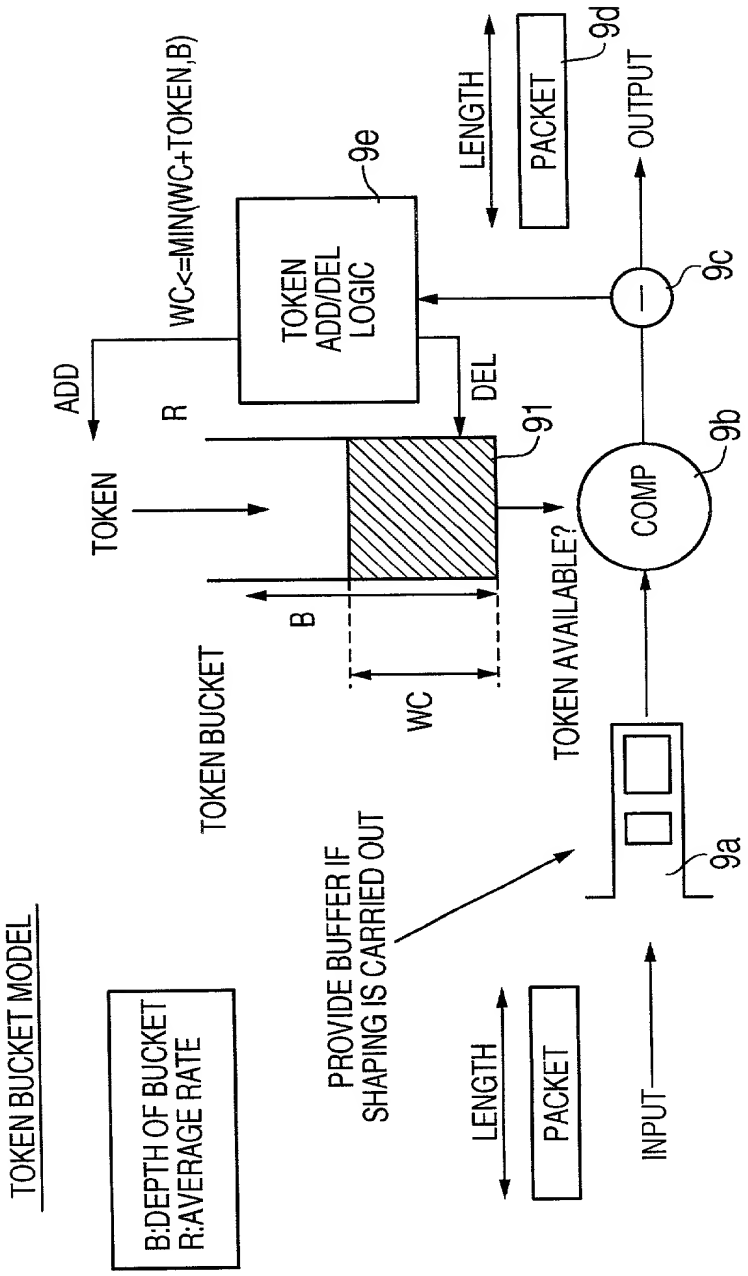
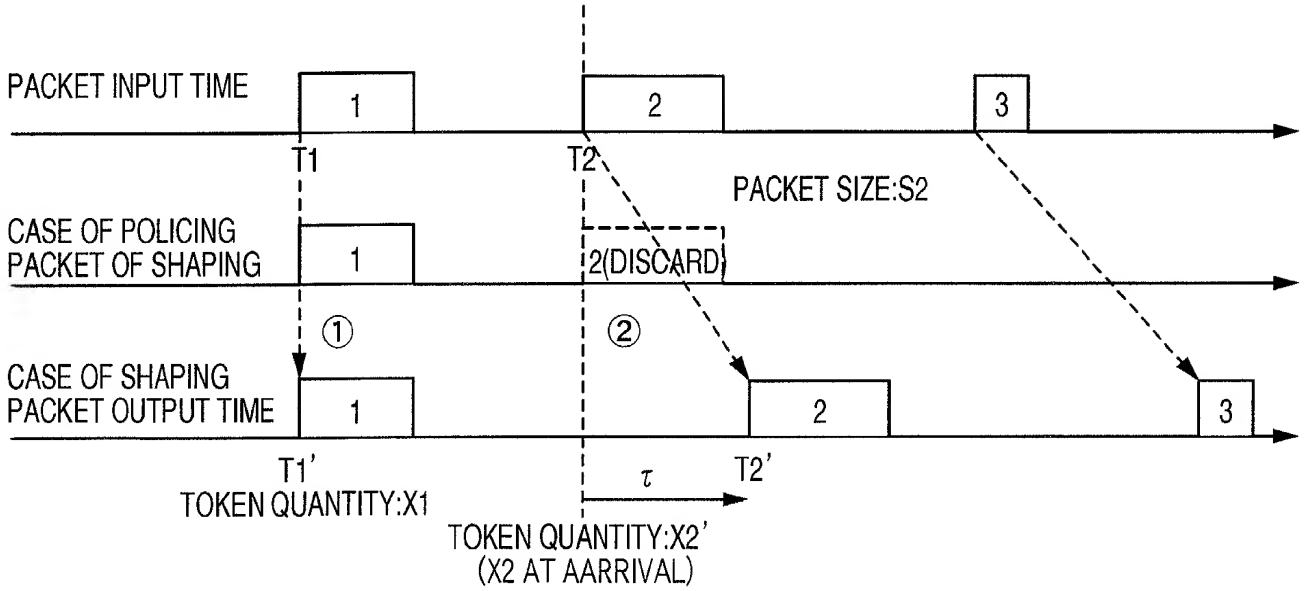

TOKEN BUCKET MODEL
B:DEPTH OF BUCKET
R:AVERAGE RATE
PROVIDE BUFFER IF
SHAPING IS CARRIED OUT
TOKEN BUCKET
TOKEN
ADD
R
WC<=MIN(WC+TOKEN,B)
TOKEN
ADD/DEL
LOGIC
DEL
9e
91
B
WC
TOKEN AVAILABLE?
COMP
9b
LENGTH
PACKET
9d
OUTPUT
9c
INPUT
LENGTH
PACKET
9a

Fig.16



1)TOKEN QUANTITIY AT TIME T2:X2=x1+(T2-T1)*R
LACK OF TOKEN,IF S2>X2
2)POLICING
IMMEDIATELY DISCARD
3)SHAPING
NOT LACK OF TOKEN,IF PACKET IS TRANSMITTED AT TIME
(τ+T2)(S2=X1+{(τ+T2)-T1}*R).PACKET IS TRANSMITTED WITH DELAY τ